

## LISTING OF CLAIMS

1. (Currently Amended) A method, comprising:
  - building a steganographic extractor to extract hidden information contained in binary images;
  - loading the steganographic extractor during a pre-boot phase of a computer system;
  - discovering, during the pre-boot phase, binary firmware images on which a steganographic operation has been performed to generate hidden information contained within each binary firmware image; and
  - extracting, via the steganographic extractor during the pre-boot phase, the hidden information contained in each binary firmware image ~~that is discovered;~~
    - obtaining a digital signature contained in the hidden information that is extracted from one of the binary firmware images;
    - comparing the digital signature with a known authentic digital signature to determine an authenticity of that binary firmware image;
    - providing a notification message to at least one of a user and a system environment log indicating the binary firmware image could not be authenticated;
    - providing the user with an option to load the binary firmware image that could not be authenticated; and
    - loading the binary firmware image in response to the request by the user.
2. (Cancelled).
3. (Currently Amended) The method of claim 2, further comprising:
  - loading the binary firmware image if it is determined that the digital signature that is extracted matches the known authentic digital signature.
4. (Cancelled).

5. (Cancelled).

6. (Currently Amended) The method of claim 1, wherein the extracted hidden information ~~that is~~ extracted contains at least one of manufacturer and versioning information, the method further comprising:

obtaining said at least one of manufacturer and versioning information from the hidden information; and

storing said at least one of manufacturer and versioning information in an asset management log.

7. (Original) The method of claim 1, further comprising:

discovering both binary firmware images having hidden information and binary firmware images not having hidden information during the pre-boot phase;

determine whether each of the binary firmware images is a trusted type or a non-trusted type; and

directly loading a binary firmware image if it is determined to be a trusted type.

8. (Currently Amended) The method of claim [[7]] 1, wherein the binary firmware images discovered in a boot firmware device are deemed to be of trusted type, while binary firmware images stored in firmware storage that is external to the boot firmware device are deemed to be of non-trusted type.

9. (Currently Amended) The method of claim 1, wherein the steganographic extractor includes a replacement map that maps equivalent sequences of op codes to one another, with one of the equivalent op code sequences assigned a "0" state and the other assigned a "1" state, the method further comprising:

parsing a binary firmware image to identify replaceable op code sequences defined by the replacement map;  
for each replaceable op code sequence found,  
determining whether it corresponds to a "0" or a "1" state; and  
adding the "0" or "1" state to an extracted binary string ~~that is used to hold for~~  
holding the extracted hidden information that is extracted.

10. (Original) The method of claim 9, wherein the binary firmware image is parsed backward, beginning with the end of the image.

11. (Original) The method of claim 9, wherein the extracted binary string contains encrypted information, the method further comprising decrypting the extracted binary string.

12. (Original) The method of claim 9, further comprising converting the extracted binary string into an alphanumeric form.

13-20. (Canceled).

21. (Currently Amended) A machine-readable computer-readable storage medium to provide instructions, which when executed perform operations, including:  
loading a steganographic extractor during a pre-boot phase of a computer system;  
discovering, during the pre-boot phase, binary firmware images on which a steganographic operation has been performed to generate hidden information embedded within each binary firmware image; ~~and~~  
extracting, via the steganographic extractor during the pre-boot phase, the hidden information contained in each binary firmware image ~~that is discovered;~~  
obtaining a digital signature contained in the hidden information that is extracted from one of the binary firmware images;

comparing the digital signature with a known authentic digital signature to determine an authenticity of that binary firmware image;

providing a notification message to at least one of a user and a system environment log indicating the binary firmware image could not be authenticated;

providing the user with an option to load the binary firmware image that could not be authenticated; and

loading the binary firmware image in response to the request by the user.

22. **(Currently Amended)** The ~~machine-readable~~ computer-readable storage medium of claim 21, wherein the media further includes instructions embodied as the steganographic extractor.

23. **(Cancelled).**

24. **(Currently Amended)** The ~~machine-readable~~ computer-readable storage medium of claim 21, further comprising instructions for performing the operations comprising of:

determining op code sequences that are identifiable to the steganographic extractor as representing steganographic data;

performing a hash on a portion of a binary firmware image that includes op codes that are exclusive of the op code sequences that represent steganographic data to obtain an image hash digest;

extracting an authenticated hash digest from the hidden information; and

comparing the image hash digest to the authenticated hash digest to determine an authenticity of the binary firmware image.

25. **(Currently Amended)** The ~~machine-readable~~ computer-readable storage medium of claim ~~[[21]]~~ 24, further comprising instructions for performing the operations comprising of:

retrieving a decryption key; and  
decrypting the hidden information that is extracted with the decryption key to  
obtain the authenticated hash digest.

26. (Currently Amended) The ~~machine-readable~~ computer-readable storage medium of claim 21, wherein the hidden information includes asset management information, and the ~~machine-readable~~ computer-readable storage medium further includes instructions for performing the operations comprising of:

retrieving the asset management information from the hidden information that is extracted; and  
storing the asset management information that is retrieved.

27. (Currently Amended) A system comprising:

a processor;  
a memory, coupled to with the processor;  
a flash device, coupled to with the processor and having firmware instructions stored therein, which when executed perform operations including:

loading a steganographic extractor during a pre-boot phase of a computer system into the memory;

discovering, during the pre-boot phase, binary firmware drivers containing hidden information that was embedded via a steganographic operation; ~~and~~

extracting, via the steganographic extractor during the pre-boot phase, the hidden information contained in each binary firmware ~~image driver~~ that is discovered.

determining whether each of the binary firmware drivers is a trusted type or a non-trusted type, and

directly loading the binary firmware driver which is determined to be of the trusted type.

28. (Original) The system of claim 27, wherein the flash device further includes firmware instructions embodied as the steganographic extractor.

29. (Original) The system of claim 27, wherein the flash device further includes firmware instructions for performing the operations comprising of:

- determining op code sequences that are mapped to identify steganographic data;
- performing a hash on a portion of a binary firmware driver image that includes op codes that are exclusive of the op code sequences that are mapped to identify steganographic data to obtain an image hash digest;
- extracting an authenticated hash digest from the hidden information; and
- comparing the image hash digest to the authenticated hash digest to determine an authenticity of the binary firmware image.

30. (Currently Amended) The system of claim 29, wherein the hidden information ~~was is~~ encrypted ~~using via~~ a private key and wherein the flash device further includes firmware instructions for performing the operations comprising of:

- retrieving a public key stored in the computer system corresponding to the private key; and
- employing the public key to decrypt the hidden information that is extracted from one of the binary firmware drivers.